

Errata Sheet

August 12, 2005 / Release 1.1

Device: SAK-C167CS-LM
SAK-C167CS-L33M
SAK-C167CS-L40M
SAK-C167CS-4RM
SAK-C167CS-4R33M
SAK-C167CS-4R40M

Stepping Code / Marking: ES-CA, CA, CA+

Package: P-MQFP-144-8

This Errata Sheet describes the deviations from the current user documentation. The classification and numbering system is module oriented in a continual ascending sequence over several derivatives, as well already solved deviations are included. So gaps inside this enumeration could occur.

The current documentation is:

- C167CS Data Sheet V2.2 2001-08
- C167CS User's Manual V2.0 2000-07
- Instruction Set Manual V2.0, 2001-03

Hyperlink (follow link to Documents):

<http://www.infineon.com/C167CS>
<http://www.infineon.com/C167CS>
<http://www.infineon.com/c166-family>

Note: *Devices marked with EES or ES are engineering samples which may not be completely tested in all functional and electrical characteristics, therefore they should be used for evaluation only.*

The specific test conditions for EES and ES are documented in a separate Status Sheet.

Change summary from Errata Sheet Rel.1.0 for C167CS-4RM/LM devices with stepping code/marking ES-CA, CA to this Errata Sheet Rel.1.1 for C167CS-4RM/LM devices with stepping code/marking ES-CA, CA, CA+:

- Device step CA+ included
- Data Transmission in Slave Mode (SSC.9) included

- Edge Selection for Capture Function if CT3 = 1 and CI = 01b or 10b (T5CON.D1, Documentation Update)
- Write access to registers PWx and PPx while bit PTRx = 0 (PWM.D1, Documentation Update)
- ID-Registers included (Documentation Update)
- Links to Infineon internet pages updated (status August 5, 2005)

Functional Problems:

PWRDN.1: Execution of PWRDN Instruction while pin NMI# = high

When instruction PWRDN is executed while pin NMI# is at a high level, power down mode should not be entered, and the PWRDN instruction should be ignored. However, under the conditions described below, the PWRDN instruction may not be ignored, and no further instructions are fetched from external memory, i.e. the CPU is in a quasi-idle state. This problem will only occur in the following situations:

- a) the instructions following the PWRDN instruction are located in external memory, and a **multiplexed bus** configuration **with memory tristate waitstate** (bit MTTCx = 0) is used, or
- b) the instruction preceding the PWRDN instruction **writes** to external memory or an XPeripheral (XRAM, CAN), and the instructions following the PWRDN instruction are located in external memory. In this case, the problem will occur for any bus configuration.

Note: the on-chip peripherals are still working correctly, in particular the Watchdog Timer will reset the device upon an overflow. Interrupts and PEC transfers, however, can not be processed. In case NMI# is asserted low while the device is in this quasi-idle state, power down mode is entered.

Workaround:

Ensure that no instruction which writes to external memory or an XPeripheral precedes the PWRDN instruction, otherwise insert e.g. a NOP instruction in front of PWRDN. When a multiplexed bus with memory tristate waitstate is used, the PWRDN instruction should be executed out of internal RAM or XRAM.

ADC.11: Modifications of ADM field while bit ADST = 0

The A/D converter may unintentionally start one auto scan single conversion sequence when the following sequence of conditions is true:

- (1) the A/D converter has finished a fixed channel single conversion of an analog channel $n > 0$ (i.e. contents of ADCON.ADCH = n during this conversion)
- (2) the A/D converter is idle (i.e. ADBSY = 0)
- (3) then the conversion mode in the ADC Mode Selection field ADM is changed to Auto Scan Single (ADM = 10b) or Continuous (ADM = 11b) mode without setting bit ADST = 1 with the same instruction

Under these conditions, the A/D converter will unintentionally start one auto scan single conversion sequence, beginning with channel n-1, down to channel number 0.

In case the channel number ADCH has been changed before or with the same instruction which selected the auto scan mode, this channel number has no effect on the unintended auto scan sequence (i.e. it is not used in this auto scan sequence).

Note:

When a conversion is already in progress, and then the configuration in register ADCON is changed,

- the new conversion mode in ADM is evaluated after the current conversion
- the new channel number in ADCH and new status of bit ADST are evaluated after the current conversion when a conversion in fixed channel conversion mode is in progress, and after the current conversion sequence (i.e. after conversion of channel 0) when a conversion in an auto scan mode is in progress.

In this case, it is a specified operational behaviour that channels n-1 .. 0 are converted when ADM is changed to an auto scan mode while a fixed channel conversion of channel n is in progress (see e.g. C167CS User's Manual, V2.0, p18-6)

Workaround:

When an auto scan conversion is to be performed, always start the A/D converter with the same instruction which sets the configuration in register ADCON.

CAPCOM.4: SW Access to P1H Overwrites CAPCOM HW Settings

HW settings on P1H.7...4 by CAPCOM compare output functions (CC24 ... CC27) can be overwritten by SW accesses to P1H.7...0 on the same port.

Read modify write operations like BSET, BFLDx, OR, ... read the input or output latches respectively, modify the affected bits and write back the result to the output latches of the whole port (P1H.7...0).

In case a compare event has occurred after the read phase, but before the write-back phase of such an instruction, the output signal change of the compare event is lost or only a short pulse (≥ 1 TCL) may appear. The bit protection mechanism to avoid these effects is out of function on P1H.

Workarounds:

1. Avoid the combination of HW and SW write accesses to P1H. HW access only to CC24 ... CC27 or SW access only to P1H.7 ... 0 works properly.
2. Use „interrupt only“ compare modes (CCMODx = 100 or 110) and modify the port pin in the interrupt service routine by software to avoid the combination of HW and SW accesses.

SSC.9: Data Transmission in Slave Mode

During data reception in slave mode of the SSC module, sporadically the shift clock supplied by the external master on pin SCLK may not be properly recognized due to a synchronization problem when all of the following conditions are true:

- (1) the latching edge for the serial data is the **falling** edge of SCLK (i.e. both bit SSCPO = 1 and bit SSCPH = 1, or SSCPO = 0 and SSCPH = 0 in register SSCCON), and
- (2) the transmit buffer **SSCTB** of the slave has **not** been **written** prior to the start of the reception (initiated by the master asserting the shift clock SCLK), and
- (3) a specific time window (phase delay) is hit by the serial shift clock SCLK in relation to the internal system clock of the slave. Therefore, this synchronization problem will occur in particular when the slave device is clocked (on XTAL1) by an external clock generation circuit which is independent from the clock generation circuit of the master (i.e. slave and master clocks are **asynchronous**).

When the problem occurs, this results in missing bits in the character received in SSCTB, and in duplicated bits in the character transmitted on pin MRST of the slave. As a consequence, interrupt generation in the slave is delayed by the number of missed bits.

Workaround

For systems using the falling edge of SCLK as latching edge (see condition (1) above), always write to the transmit buffer SSCTB prior to any reception in slave mode of the SSC module. For the second and all following characters, e.g. write a (dummy) character to SSCTB in the receive interrupt routine, or use a PEC transfer triggered by the transmit interrupt request to write to SSCTB. In this case, the critical synchronization path is not used, and the problem will not occur.

X9: Read Access to XPERs in Visible Mode

The data of a read access to an XBUS-Peripheral (XRAM, CAN) in Visible Mode is not driven to the external bus. PORT0 is tristated during such read accesses.

Note that in Visible Mode PORT1 will drive the address for an access to an XBUS-Peripheral, even when only a multiplexed external bus is enabled.

X17: XBUS Access after External Access with EWEN = 1 and/or BSWC = 1

1. When the last external bus access which precedes an internal XBUS-Peripheral access is performed with bit **BUSCONx.11/BSWCx = 1 (BUSCON Switch Tristate Waitstate)**, the XBUS cycle will be performed without BUSCON Switch Tristate Waitstate, and the BUSCON Switch Tristate Waitstate will be inserted at the beginning of the next external bus cycle.

Workaround: the data of an internal XBUS write cycle will be driven on Port 0 starting 2 TCL after the rising edge of ALE. In case the tdf (output disable time) specification of the external device does not meet the following requirements

$tdf < 4 \text{ TCL} - 15 + t_f \text{ [ns]}$ for a multiplexed bus cycle via BUSCONx, or
 $tdf < 2 \text{ TCL} - 15 + t_f \text{ [ns]}$ for a non-multiplexed bus cycle via BUSCONx,

then use a standard Memory Tristate Time Waitstate instead of a BUSCON Switch Tristate Waitstate, i.e. set bit BUSCONx.5/MTTCx = 0.

2. When the last external bus access which precedes an internal XBUS-Peripheral access is performed with bit **BUSCONx.8/EWENx = 1 (Early Write)**, the XBUS cycle will also be performed with 'Early Write'.

Workaround: not required, internal XBUS-Peripherals tolerate write accesses with Early Write.

CPU.22: Z Flag after PUSH and PCALL

The Z flag in the PSW is erroneously set to '1' by **PUSH reg** or **PCALL reg, rel** instructions when all of the following conditions are true:

- (a) for **PUSH reg** instructions:
 - the contents of the high byte of the GPR or (E)SFR which is pushed is 00h, and
 - the contents of the low byte of the GPR or (E)SFR which is pushed is > 00h, and
 - the contents of GPR Rx is odd, where x = 4 msbs of the 8-bit 'reg' address of the pushed GPR or (E)SFR

Examples:

PUSH R1 (coding: **F1** EC): incorrect setting of Z flag if contents of **R15** is odd,
and $00FFh \geq \text{contents of R1} \geq 0001h$
PUSH DPP3 (coding: **03** EC): incorrect setting of Z flag if contents of **R0** is odd,
and $00FFh \geq \text{contents of DPP3} \geq 0001h$

(b) for **PCALL reg, rel** instructions:

- when the contents of the high byte of the GPR or (E)SFR which is pushed is 00h, and
- when the contents of the low byte of the GPR or (E)SFR which is pushed is odd

This may lead to wrong results of instructions following PUSH or PCALL if those instructions explicitly (e.g. BMOV .., Z; JB Z, ..; ..) or implicitly (e.g. JMP cc_Z, ..; JMP cc_NET, ..; ..) evaluate the status of the Z flag before it is newly updated.

Note that some instructions (e.g. CALL, ..) have no effect on the status flags, such that the status of the Z flag remains incorrect after a PUSH/PCALL instruction until an instruction that correctly updates the Z flag is executed.

Example:

```
PUSH R1          ; incorrect setting of Z flag if R15 is odd
CALL proc_xyz    ; Z flag remains unchanged (is a parameter for proc_xyz)
...
proc_xyz:
  JMP cc_Z,end_xyz ; Z flag evaluated with incorrect setting
...
end_xyz:
```

Effect on Tools:

The **Hightec** C166 tools (all versions) don't use the combination of PUSH/PCALL and the evaluation of the Z flag. Therefore, these tools are not affected.

The code generated by the **Keil** C166 Compiler evaluates the Z flag only after MOV, CMP, arithmetic, or logical instructions. It is never evaluated after a PUSH instruction. PCALL instructions are not generated by the C166 Compiler.

This has been checked with all C166 V3.xx and V4.xx compiler versions. Even the upcoming V5.xx is not affected by the CPU.22 problem.

The assembler portions of the C166 V3.xx and V4.xx Run-Time Libraries, the RTX166 Full and TX166 Tiny Real Time Operating system do also not contain any evaluation of the Z flag after PUSH or PCALL.

The **TASKING** compiler V7.5r2 never generates a PCALL instruction, nor is it used in the libraries. The PUSH instruction is only used in the entry of an interrupt frame, and sometimes on exit of normal functions. The zero flag is not a parameter or return value, so this does not give any problems.

Previous versions of TASKING tools: V3.x and higher are not affected, versions before 3.x are most likely not affected. Contact TASKING when using versions before V3.x.

Since code generated by the C166 **compiler** versions mentioned before is **not** affected, analysis and workarounds are **only** required for program parts written in **assembler**, or instruction sequences inserted via inline assembly.

Workaround (for program parts written in assembler):

Do not evaluate the status of the Z flag generated by a PUSH or PCALL instruction. Instead, insert an instruction that correctly updates the PSW flags, e.g.

```
PUSH reg
CMP reg, #0          ; updates PSW flags
                    ; note: CMP additionally modifies the C and V flags,
                    ; while PUSH or MOV leaves them unaffected
JMPR cc_Z, label_1 ; implicitly tests Z flag

or
PCALL reg, procedure_1
...
procedure_1:
MOV ONES, reg        ; updates PSW flags
JMPR cc_NET, label_1 ; implicitly tests flags Z and E
```

Hints for Detection of Critical Instruction Combinations

Whether or not an instruction following *PUSH reg* or *PCALL reg, rel* actually causes a problem depends on the program context. In most cases, it will be sufficient to just analyze the instruction following PUSH or PCALL. In case of PCALL, this is the instruction at the call target address.

- Support Tool for Analysis of Hex Files

For complex software projects, where a large number of assembler source (or list) files would have to be analyzed, Infineon provides a tool **aiScan22** which scans hex files for critical instruction sequences and outputs diagnostic information. This tool (description + software) is available as part of the Application Note **ap1679** 'Scanning for Problem CPU.22' in the Microcontroller section of the internet pages of Infineon Technologies:

<http://www.infineon.com/c166-family> (follow link to Application Notes – 16-bit Microcontrollers)

- Individual Analysis of Assembler Source Code

With respect to problem CPU.22, all instructions of the C166 instruction set can be classified into the following groups:

- **Arithmetic/logic/data movement** instructions as successors of PUSH/PCALL (correctly) modify the condition flags in the PSW according to the result of the operation.
- These instructions may **only** cause a problem if the **PSW** is a source or source/destination operand:
ADD/B, ADDC/B, CMP/B, CMPD1/2, CMPI1/2, SUB/B, SUBC/B
AND/B, OR/B, XOR/B
ASHR
MOV/B, MOV/BZ/MOVBZ
SCXT
PUSH, PCALL ; analysis must be repeated for successor of PUSH/PCALL
- The following instructions (most of them with immediate or register (Rx) addressing modes) can **never** cause a problem when they are successors of PUSH/PCALL:
CPL/B, NEG/B
DIV/U, DIVL/U, MUL/U
SHL/SHR, ROL/ROR, PRIOR

POP	
RETI	; updates complete PSW with stacked value
RETP	; updates condition flags
PWRDN	; program restarts after reset
SRST	; program restarts

- **Conditional branch instructions** which may evaluate the Z flag as successors of PUSH/PCALL:

JB/JNB Z, *rel* ; directly evaluates Z flag
CALLA/CALLI, JMPA/JMPI/JMPR with the following condition codes
cc_Z, cc_EQ, cc_NZ, cc_NE
cc_ULE, cc_UGT, cc_SLE, cc_SGT
cc_NET

- For these branch conditions, the branch may be performed in the wrong way.
- For other branch conditions, the branch target as well as the linear successor of the branch instruction must be analyzed (since these branch instruction don't modify the PSW flags).

- For **instructions that have no effect on the condition flags** and that don't evaluate the Z flag, the instruction that follows this instruction must be analyzed. These instructions are

NOP	
ATOMIC, EXTxx	
DISWDT, EINIT, IDLE, SRVWDT	
CALLR, CALLS, JMPS	; branch target must be analyzed
RET, RETS	; return target must be analyzed (value pushed by PUSH/PCALL = return IP,
	; Z flag contains information whether intra-segment target address = 0000h or not)
TRAP	; both trap target and linear successor must be analyzed, since Z flag may be
	; incorrect in PSW on stack as well as in PSW at entry of trap routine

- For **bit modification instructions**, the problem may only occur if a source bit is the Z flag, and/or the destination bit is in the PSW, but not the Z flag. These instructions are:

BMOV/BMOVN	
BAND/BOR/BXOR	
BCMP	
BFLDH	
BFLDL	; problem only if bit 3 of @@ mask = 0, i.e. if Z is not selected
BCLR/BSET	; problem only if operand is not Z flag
JBC/JNBS	; wrong branch if operand is Z flag

CAN.7 Unexpected remote frame transmission

The on-chip CAN module may send an unexpected remote frame with the identifier=0, when a pending transmit request of a message object is disabled by software.

Detailed Description

There are three possibilities to disable a pending transmit request of a message object (n=1..14):

- Set CPUUPDn element
- Reset TXROn element

- Reset MSGVALn element

Either of these actions will prevent further transmissions of message object n.

The symptom described above occurs when the CPU accesses CPUUPD, TXRQ or MSGVAL, while the pending transmit request of the corresponding message object is transferred to the CAN state machine (just before start of frame transmission). At this particular time the transmit request is transferred to the CAN state machine before the CPU prevents transmission. In this case the transmit request is still accepted from the CAN state machine. However the transfer of the identifier, the data length code and the data of the corresponding message object is prevented. Then the pre-charge values of the internal “hidden buffer” are transmitted instead, this causes a remote frame transmission with identifier=0 (11 bit) and data length code=0.

This behavior occurs only when the transmit request of message object n is pending and the transmit requests of other message objects are **not** active (single transmit request). If this remote frame loses arbitration (to a data frame with identifier=0) or if it is disturbed by an error frame, it is **not** retransmitted.

Effects to other CAN nodes in the network

The effect leads to delays of other pending messages in the CAN network due to the high priority of the Remote Frame. Furthermore the unexpected remote frame can trigger other data frames depending on the CAN node's configuration.

Workarounds

1. The behavior can be avoided if a message object is not updated by software when a transmission of the corresponding message object is pending (TXRQ element is set) **and** the CAN module is active (INIT = 0). If a re-transmission of a message (e.g. after lost arbitration or after the occurrence of an error frame) needs to be cancelled, the TXRQ element should be cleared by software as soon as NEWDAT is reset from the CAN module.
2. The nodes in the CAN system ignore the remote frame with the identifier=0 and **no** data frame is triggered by this remote frame.

Deviations from Electrical- and Timing Specification:

The following table lists the deviations of the DC/AC characteristics from the specification in the C167CS Data Sheet V2.2 2001-08:

Problem short name	Parameter	Symbol	Limit Values		Unit	Test Condition
			min.	max.		
DC.HYS.250	Input Hysteresis (Special Threshold)	HYS	250 instead of 400	-	mV	
DC.VDD.4	Digital Supply Voltage	V _{DD}	4.7 instead of 4.5	5.5	V	125 °C C167CS-x40M only

A/D Converter Characteristics:

ADCC.2.5: ADC Overload Current

During exceptional conditions in the application system an overload current I_{OV} can occur on the analog inputs of the A/D converter when $V_{AIN} > V_{DD}$ or $V_{AIN} < V_{SS}$. For this case, the following conditions are specified in the Data Sheet:

$$I_{OVmax} = | \pm 5 \text{ mA} |$$

The specified total unadjusted error $TUE_{max} = | \pm 2 \text{ LSB} |$ @ Port 5 and $TUE_{max} = | \pm 10 \text{ LSB} |$ @ PORT1 is guaranteed only if the absolute sum of input overload currents on all analog input pins does not exceed 10 mA.

Due to an internal problem, the specified TUE value is only met for a **positive** overload current $0 \text{ mA} \leq I_{OV} \leq +5 \text{ mA}$ (all currents flowing into the microcontroller are defined as positive and all currents flowing out of it are defined as negative).

If the exceptional conditions in the application system cause a **negative** overload current, then the maximum TUE can be exceeded (depending on value of I_{OV} , R_{AIN} , R_{AREF} and R_{AGND}):

Port 5 Problem Description in Detail:

1. Overload Current at Analog channel AN0 - AN15

If a **negative** overload current I_{OV} occurs on analog input channel ANn, then an additional current I_{AN} (crosstalk current) is caused at the neighbour channels ANn-1 and ANn+1. This behavior causes an additional unadjusted error AUE to the ADC result.

Relation between I_{AN} and I_{OV} :

$$I_{ANn+1} = \text{ovf-1} * I_{OVn}$$

$$I_{ANn-1} = \text{ovf-1} * I_{OVn}$$

2. Overload Current at Digital channel P7.7

A negative overload current I_{OV} at digital Port P7.7 causes an additional current I_{AN} at the analog input AN0. The relation between both channels is also defined by ovf-1:

$$I_{AN0} = \text{ovf-1} * I_{OVP7.7}$$

3. Overload Current at Analog channel AN12 – AN15 and influence to V_{AREF}

If an overload current I_{OV} occurs on analog input channel AN12 – AN15 then an additional current I_{AREF} (crosstalk current) is caused at pin V_{AREF} .

Depending on R_{AREF} , the internal resistance of the reference voltage, the crosstalk current I_{AREF} at pin V_{AREF} can cause an additional unadjusted error AUE to all other analog channels.

In case $R_{AREF} \leq 2450 \text{ Ohm}$ [$R_{AREF} \leq ((\text{LSB}/2) / (I_{OVmax} * \text{ovf-3})$] the maximum possible additional error to all other channels is smaller than 0.5 LSB with the condition of $I_{OVmax} = |\pm 5 \text{ mA}|$ at AN12 - AN15.

Relation between I_{AREF} and I_{OV} :

$$I_{AREF} = \text{ovf-3} * I_{OVn} \quad (12 \leq n \leq 15)$$

Note: The influence to the reference voltage V_{AREF} caused by I_{OVn} (shift of V_{AREF}) is maximum for $V_{AINn} = V_{AREF}$ and the influence is minimum for $V_{AINn} = 0V$. The condition $R_{AREF} \leq 2450 \text{ Ohm}$ and 0.5 LSB is calculated for the worst case at $V_{AINn} = V_{AREF}$.

4. Values of ovf-1 and ovf-3

Parameter	Symbol	Min	Max
Overload factor-1	ovf-1	- 0.0001	0
Overload factor-3	ovf-3	- 0.0002	0

These Values are the absolute **maximum values measured in the lab and not tested!**

5. Effects on the Conversion Result and TUE

The effect on the conversion result and the TUE has to be calculated based on the crosstalk current and the impedance of the analog source $R_{ASRC} \cdot I_{ANn}$ causes an external voltage $U_{\Delta n}$ at the analog channel ANn (same principle for V_{AREF} and V_{AGND}) which is the reason for an additional unadjusted error **AUE** of the conversion result. This AUE can increase the specified total unadjusted error TUE (Specified value: TUE = $\pm 2 \text{ LSB}$). The voltage $U_{\Delta n}$ is nearly independent on the voltage value of the analog source.

$$\begin{aligned} U_{\Delta n} &= I_{ANn} * R_{ASRC} \\ U_{\Delta AREF} &= I_{AREF} * R_{AREF} \\ U_{\Delta AGND} &= I_{AGND} * R_{AGND} \\ AUE &= U_{\Delta n} / 1 \text{ LSB} \\ TUE &= (\pm 2 \text{ LSB}) \pm AUE \end{aligned} \quad [U_{\Delta n} \text{ in mV and LSB in mV}]$$

Note: A negative overload current I_{OVn} decreases the analog signal voltage V_{AINn} and causes a negative AUE.

6. Calculation Example

Assumed system values:

$$\begin{aligned} I_{OV4} &= -1 \text{ mA} && \text{negative overload current at P5.4} \\ R_{ASRC} &= 20 \text{ kOhm} && \text{resistance of the external sensor at P5.5} \\ V_{AREF} &= 5 \text{ V} && 1 \text{ LSB} = 4.9 \text{ mV} \\ R_{AREF} &\text{ has to be considered for } I_{OV} \text{ at ANn } (12 \leq n \leq 15) \\ R_{AREF} &= 500 \text{ Ohm} && \rightarrow V_{AREF} \text{ shift error is negligible} \end{aligned}$$

$$\begin{aligned}
I_{AN5} &= \text{ovf1} * I_{OV4} \\
I_{AN5} &= (-1.0 \times 10^{-4}) * (-1 \text{ mA}) \\
I_{AN5} &= 0.10 \mu\text{A}
\end{aligned}$$

$$\begin{aligned}
U_{\Delta 5} &= I_{AN5} * R_{ASRC} \\
U_{\Delta 5} &= 0.10 \mu\text{A} * 20 \text{ k}\Omega \\
U_{\Delta 5} &= 2.0 \text{ mV}
\end{aligned}$$

$$\begin{aligned}
AUE &= U_{\Delta n} / 1 \text{ LSB} \\
AUE &= 2.0 \text{ mV} / 4.9 \text{ mV} \\
AUE &= 0.41 \text{ LSB}
\end{aligned}$$

Result: The negative overload current I_{OV4} of this system example can distort the real result of AN5 by an additional unadjusted error, $0 \text{ LSB} \leq AUE \leq 0.41 \text{ LSB}$.
The TUE is in the range of $-2.41 \text{ LSB} \leq TUE \leq 2.0 \text{ LSB}$.

PORT1 Problem Description in Detail:

1. Overload Current at analog channel AN16 – AN23

If a **negative** overload current I_{OV} occurs on analog input channel ANn, then an additional current I_{AN} (crosstalk current) is caused at the neighbour channels ANn-1 and ANn+1. This behavior causes an additional unadjusted error AUE to the ADC result.

Relation between I_{AN} and I_{OV} :

$$\begin{aligned}
I_{ANn+1} &= \text{ovf-11} * I_{OVn} \\
I_{ANn-1} &= \text{ovf-11} * I_{OVn}
\end{aligned}$$

2. Overload Current at digital channel P0H.7

A negative overload current I_{OV} at digital Port P0H.7 causes an additional current I_{AN} at the analog input AN16. The relation between both channels is also defined by ovf-11:

$$I_{AN0} = \text{ovf-11} * I_{OVP0H.7}$$

3. Values of ovf-11

Parameter	Symbol	Min	Max
Overload factor-11	ovf-11	- 0.0008	0

This value is the absolute **maximum value measured in the lab and is not tested!**

History List (since C167CS-4RM/LM device step (E)ES-AA)

Functional Problems

Functional Problem	Short Description	Fixed in step
PWRDN.1	Execution of PWRDN Instruction while pin NMI# = high	
CAPCOM.4	SW Access to P1H Overwrites CAPCOM HW Settings	
ADC.11	Modifications of ADM field while bit ADST = 0	
SSC.9	Data Transmission in Slave Mode	
CAN.7	Unexpected Remote Frame Transmission	
CPU.22	Z Flag after PUSH and PCALL	
X9	Read Access to XPERs in Visible Mode	
X17	XBUS Access after External Access with EWEN = 1 and/or BSWC = 1	
SYSCON1.1	Reset Value in Register SYSCON1	ES-BA
(POWER.19	SW- or WDT-Reset while SDD mode is active	see RST.9)
POWER.16	SDD and Sleep Mode cannot be terminated	ES-AC
POWER.15	Wake Up from Sleep Mode not possible in Prescaler Mode	ES-CA
POWER.10	Sleep Mode: CAN and ADC Modules enabled	ES-AB
POWER.8	Termination of Sleep Mode with HW Reset	(E)ES-AA
BUS.18	PEC transfers after JMPR	(E)ES-AA
BUS.19	Unlatched Chip Selects at Entry into Hold Mode	ES-BA
CPU.21	BFLDL/BFLDH Instructions after Write Operations to internal IRAM	ES-CA
RST.15.2	New Behaviour: Software/Watchdog/Short HW reset during Slow Down operation with PLL off (not for Ax-steps)	ES-CA
RST.13	Power-up with Missing Clock	ES-AB
RST.11	Missing clock at XTAL1	ES-AB
RST.9	Software/Watchdog/Short HW reset during Slow Down operation with PLL off (Ax-steps only, see also RST.15.2)	ES-BA
RST.8	Clock failure detection during external reset	ES-BA
RST.7	Latching of P0 Reset Configuration when pin EA# = high	(E)ES-AA
RST.5	HW Reset not detected in PLL Mode during Slow Down operation when PLL is switched off	(E)ES-AA
TCOMP.2	Pad Driver Temperature Compensation does not work properly (AA-steps)	ES-AB
TCOMP.3	Pad Driver Temperature Compensation disabled (AB/AC-steps) - see also note about documentation update	ES-BA
CAN.9	Contents of Message Objects and Mask of Last Message Registers after Reset	ES-CA

AC/DC Deviations

AC/DC Deviation	Short Description	Fixed in step
AC.t34.4	t34 (min): CLKOUT rising to ALE falling edge relaxed to -2ns - included in specification for tc11: -3 .. +6 ns	see Data Sheet
DC.VILS.2	Input low voltage (special threshold) V_{ILS} (max) relaxed to 1.8 V	ES-BA
DC.HYS.2	Hysteresis (special threshold) relaxed to 200 mV (Ax-steps only)	ES-BA
DC.HYS.250	Hysteresis (special threshold) relaxed to 250 mV (BA- and CA-steps)	
AC.PLL.1	PLL Jitter > 3.8 % for fcpu < 12 MHz	ES-AC
PLL.3.1	Increased PLL Jitter caused by External Access (step AB only)	ES-AC
PLL.3.2	Increased PLL Jitter caused by External Access (step AC only)	ES-BA
PLL.3.3	Increased PLL Jitter caused by External Access (step BA only)	ES-CA
DC.IIDX.1	Idle Mode Supply Current 65 mA @ 25 MHz	see DC.IID_PD.1
DC.IID_PD.1	Idle and Power Down Mode Supply Currents: not tested	see Data Sheet V2.2
ADCC.2.4	ADC Overload Current (AA-steps only)	ES-AB
ADCC.2.5	ADC Overload Current (not for AA-steps)	
DC.VDD.4	Minimum Supply Voltage VDD = 4.7V @ 125°C (C167CS-x40M CA-steps only)	

Documentation Update - Modifications of Features

(reference: C167CS User's Manual V2.0, 2000-07)

- **Registers P5DIDIS, P1DIDIS now implemented**

In the **Ax** -steps of the C167CS-4RM/LM, registers P5DIDIS and P1DIDIS are **not** implemented. They are implemented beginning with step **BA**.

- **Single Chip Mode Reset eliminated** (p. 20-20 to p. 20-24)

Note: the **Single Chip Mode Reset** feature has been eliminated beginning with the **BA**-step of the C167CS-LM/4RM: it will always read the reset configuration from PORT0, independent of the state (low or high) of pin EA#. However, the possibility to modify the configuration via register RSTCON will still be provided. See also Application Note AP1637 'Reset and System Startup Configuration via PORT0 or Register RSTCON' in the Microcontroller section of the internet pages of Infineon Technologies:

<http://www.infineon.com/c166-family> (follow link to Application Notes – 16-bit Microcontrollers)

This change ensures hardware compatibility with existing systems which are based on the C167CS-32FM up to step CB, C167CS-LM, C167CR-xx (all steps), and C167CS-4RM step BA or higher.

- **Pad Driver Temperature Compensation**

The description of the pad driver temperature compensation mechanism has been removed from User's Manual V2.0.

For systems which were using this feature before, the functionality which is implemented in the **BA**-step is described in the following:

To avoid a malfunction and additional current consumption, the on chip pad driver temperature compensation is disabled by default after reset. For this reason, the functionality of some bits has been changed:

- PTCR.3/TCE = 0 (default after reset) means that the temperature compensation sensor is disabled
- TCV = 11 (default after reset) means that the pad drivers operate with maximum driver strength

No special action is necessary if normal (strong) pad drivers shall be used. Otherwise bit field TCC may be modified and bit TCS should be set to '1' to control the pad drivers via software.

- **Package**

Beginning with step **CA**, the package has been changed to P-MQFP-144-8, see

<http://www.infineon.com/microcontrollers> (follow link to Packages - SMD - P/PG_MQFP – P-MQFP-144-1, -8)

- to C167CS User's Manual V2.0, 2000-07, p.10-30 ff.:

T5CON.D1 Edge Selection for Capture Function if CT3 = 1 and CI = 01b or 10b

In contrast to the current documentation in the User's Manuals, the edge selection for the capture function of the contents of T5 into register CAPREL triggered by transitions on pins T3IN or T3EUD works as described below. This applies to configurations where bit **CT3 = 1** and bit field **CI = 01b** or **CI = 10b**. Other functions are not affected.

Changes to the current documentation are marked in ***bold italic*** letters (see description of register T5CON in chapter 'GPT2 Auxiliary Timer T5' of the User's Manual).

Bit / Bit Field	Function
CT3	Timer 3 Capture Trigger Enable 0 : Capture trigger from pin CAPIN 1 : Capture trigger from T3 input pins
CI	Register CAPREL Capture Trigger Selection (depending on bit CT3) 0 0 : Capture disabled 0 1 : Positive transition (rising edge) on CAPIN or <i>positive transition (rising edge) on T3IN if T3EUD = 0</i> <i>negative transition (falling edge) on T3IN if T3EUD = 1</i> <i>positive transition (rising edge) on T3EUD if T3IN = 0</i> <i>negative transition (falling edge) on T3EUD if T3IN = 1</i> 1 0 : Negative transition (falling edge) on CAPIN or <i>negative transition (falling edge) on T3IN if T3EUD = 0</i> <i>positive transition (rising edge) on T3IN if T3EUD = 1</i> <i>negative transition (falling edge) on T3EUD if T3IN = 0</i> <i>positive transition (rising edge) on T3EUD if T3IN = 1</i> 1 1 : Any transition (rising or falling edge) on CAPIN or any transition on T3IN or T3EUD

The following table shows (for CT3 = 1) under which conditions a capture trigger is generated for a transition on T3IN or T3EUD, depending on the level on the respective other input T3EUD or T3IN

Level on respective other input (T3EUD or T3IN)	T3IN Input		T3EUD Input	
	Rising ↑	Falling ↓	Rising ↑	Falling ↓
High	capture if CI = 10	capture if CI = 01	capture if CI = 10	capture if CI = 01
Low	capture if CI = 01	capture if CI = 10	capture if CI = 01	capture if CI = 10

In other words, a capture trigger is generated for CI = 01 whenever the two inputs change from the same to a different logic state. A capture trigger is generated for CI = 10 whenever the two inputs change from a different to the same logic state. As an equivalent representation, a capture trigger is generated when the following logic equation is true:

CI	Capture if
01	(T3IN XOR T3EUD) = 0→1
10	(T3IN XOR T3EUD) = 1→0

Applications that require a capture event (associated with an interrupt request from register CAPREL) only for transitions on pin T3IN (independent of the status of pin T3EUD), or only for transitions on pin T3EUD (independent of the status of pin T3IN), respectively, may use the following configuration:

- externally connect pin T3IN to pin CAPIN (or pin T3EUD to pin CAPIN), respectively
- enable the capture trigger from pin CAPIN (bit CT3 = 0), and select the edge type (rising, falling, any) in bit field CI.

- to C167CS User's Manual V2.0, 2000-07, p.17-16:

PWM.D1 Write access to registers PWx and PPx while bit PTRx = 0

Clearing the timer run bit PTRx stops the associated counter PTx.

The level on the individual PWM channel outputs is controlled by comparators according to the formula:

$$\text{PWM output signal} = [\text{PTx}] \geq [\text{PWx shadow latch}].$$

While PTRx = 0, the PWx and PPx registers are transparent, i.e. a write to PWx and PPx will directly update the shadow registers as long as the corresponding timer run bit PTRx = 0.

So whenever software changes registers PTx or PWx, the respective output will reflect the condition after the change. E.g. loading timer PTx with a value greater than or equal to the value in PWx immediately sets the respective output, a PTx value below the PWx value clears the respective output.

ID-Registers

		Register:	IDMANUF	IDCHIP	IDMEM	IDPROG
Device	Step	Address:	F07Eh	F07Ch	F07Ah	F078h
C167CS-LM/4RM	BA		1820h	0C43h	1008h	0000h
	CA/CA+		1820h	0C44h	1008h	0000h

Application Hints

Handling of the SSC Busy Flag (SSCBSY)

In master mode of the High-Speed Synchronous Serial Interface (SSC), when register SSCTB has been written, flag SSCBSY is set to '1' when the baud rate generator generates the next internal clock pulse. The maximum delay between the time SSCTB has been written and flag SSCBSY=1 is up to 1/2 bit time. SSCBSY is cleared 1/2 bit time after the last latching edge.

When polling flag SSCBSY after SSCTB has been written, SSCBSY may not yet be set to '1' when it is tested for the first time (in particular at lower baud rates). Therefore, e.g. the following alternative methods are recommended:

1. test flag SSCRIR (receive interrupt request) instead of SSCBSY (in case the receive interrupt request is not serviced by CPU interrupt or PEC), e.g.
loop: BCLR SSCRIR ;clear receive interrupt request flag
 MOV SSCTB, #xyz ;send character
wait_tx_complete:
 JNB SSCRIR, wait_tx_complete ;test SSCRIR
 JB SSCBSY, wait_tx_complete ;test SSCBSY to achieve original
 timing(SSCRIR may be set 1/2 bit
 time before SSCBSY is cleared)
2. use a software semaphore bit which is set when SSCTB is written and is cleared in the SSC receive interrupt routine

Note on Interrupt Register behaviour of the CAN module

Due to the internal state machine of the CAN module, a specific delay has to be considered between resetting INTPND and reading the updated value of INTID. See Application Note AP2924 "Interrupt Register behaviour of the CAN module in 16-bit Microcontrollers" in the Microcontroller section of the internet pages of Infineon Technologies:

<http://www.infineon.com/c166-family> (follow link to Application Notes – 16-bit Microcontrollers)

Oscillator Watchdog and Prescaler Mode

The OWD replaces the missing oscillator clock signal with the PLL clock (base frequency).

- In **direct drive** mode the PLL base frequency is used directly ($f_{cpu} = 2...5$ MHz).
- In **prescaler** mode the PLL base frequency is divided by 2 ($f_{cpu} = 1...2.5$ MHz).

(see also C167CS User's Manual V2.0, p.6-8)

Maintenance of ISNC register

The RTC and PLL interrupts share one interrupt node (XP3IC). If an interrupt request occurs the request bit in the Interrupt Subnode Control register has to be checked and cleared by software. To avoid a conflict with the next hardware interrupt request it is recommended to first clear both the request and the enable bit, and then to set the enable bit again.

Example for a RTC interrupt service routine (for Tasking C compiler):

```
...
If (RTCIR)
{
    _bfld (ISNC, 0x0003, 0x0000); // clear RTCIE and RTCIR
    _putbit (1, ISNC, 1);          // set RTCIE
}
...
```

In assembly language:

```
...
EXTR    #1
JNB     RTCIR, no_rtc_request
EXTR    #2          ; no further interruption of this sequence possible
BFLDL   ISNC, #03h, #00h ; clear RTCIE and RTCIR
BSET    RTCIE          ; set RTCIE
...
no_rtc_request:
```

Type_RE Oscillator Configuration for Low Power Operation

The type_RE main oscillator is optimized for oscillation with a crystal within a frequency range of 4...40 MHz. In order to achieve minimum power consumption in power saving modes where the oscillator remains active, the size of the external components (Cx1, Cx2, Rx2) of a crystal oscillator circuit may be reduced below the values which are listed in AP2420xx 'Crystal Oscillator of the C500 and C166 Microcontroller Families' in the Microcontroller section of the internet pages of Infineon Technologies: <http://www.infineon.com/c166-family> (follow link to Application Notes – 16-bit Microcontrollers)

The recommended minimum values for the external circuits are equal to the values which are listed in AP2420xx for the Type_LP2 oscillator.

Since in the CA-step the oscillator has been adjusted to the respective technology, it is strongly recommended to measure the resulting safety factor (negative resistance method) in the final target system (layout) to ensure sufficient start-up reliability. Please refer to the limits specified by the crystal supplier.

PLL lock after temporary clock failure

When the PLL is locked and the input clock at XTAL1 is interrupted then the PLL becomes unlocked, provides the base frequency (2 ... 5 MHz) and the PLL unlock interrupt request flag is set. If the XTAL1 input clock starts oscillation again then the PLL stays in the PLL base frequency. The CPU clock source is only switched back to the XTAL1 oscillator clock after a hardware reset. This can be achieved via a normal hardware reset or via a software reset with enabled bidirectional reset. It is important that the hardware reset is at least active for 1 ms, after that time the PLL is locked in any case.

Note on Early (Unlatched) Chip Select Option

As described in the User's Manuals (e.g. C167CS User's Manual, V2.0, 2000-07, p.9-11), an early (unlatched) address chip select signal (SYSCON.CSCFG = '1') becomes active together with the address and BHE (if enabled) and remains active until the end of the current bus cycle. Early address chip select signals are not latched internally and may toggle intermediately while the address is changing.

These effects may also occur on CSx# lines which are configured as RDCSx# and/or WRCSx# signals (BUSCONx.CSRENx = 1 and/or CSWENx = 1).

The position of these transitions (spikes) is at the beginning of an external bus cycle or an internal XBUS cycle, indicated by the rising edge of signal ALE. The width of these transitions is ~ 5 ns (measured at a reference level of 2.0 V with Vdd = 5.0 V). The falling edge of the spike occurs in the same relation to RD#, WR#/WRH#/WRL# and to other CS# signals as if it was an address chip select signal with early chip select option.

When CS# lines configured as RDCS# and/or WRCS# are used e.g. as output enable (OE#) signals for external devices or as clock input for shift registers, problems might occur (temporary bus contention during data float times (may be solved by tristate wait state), unexpected shift operations, etc.). When CS# lines configured as WRCS# are used as write enable (WE#) signals for external devices or FIFOs, internal locations may be overwritten with undefined data.

When CS# lines are used as chip enable (CE#) signals for external memories, usually no problems are expected, since the falling edge of the spikes has the same characteristics as the falling edge of an access with a regular early (unlatched) address CS# signal. At this time, the memory control signals RD#, WR# (WRH#/WRL#) are on their inactive (high) levels.

Application Support Group, Munich